

I CLAIM

1. A modified LCVQ method for creating an optimal codebook; said method comprising the steps of:

5

a) initializing a base codebook B with a set of base codevectors;

b) finding the nearest neighbour for each source vector in a block of data and associating said each source vector with a base codevector in said base codebook B;

c) creating codebook C based upon the formula, $C = B \cdot W$, wherein W is a weight matrix;

10

d) updating base codebook B utilizing the following formula:

$$B = S W^T * \text{inverse}(W N W^T),$$

wherein S is a matrix where each column contains the sum of source vectors currently assigned to each base codevector in B and N is a diagonal matrix with the number of source vectors which are the closest to each base codevector in B appearing on the diagonal; and

15

e) repeating steps b) to d) until the base codevectors in base codebook B have converged.

20

2. The method of claim 1, wherein at step a) said initializing comprises randomly selecting source vectors from said block of data, without replacement.

3. The method of claim 1 wherein if at step a), said base codebook B is singular, reinitializing said base codebook B.

25

4. The method of claim 3 wherein said reinitializing comprises selecting an outlying source code vector from within said block of data and inserting said outlying source code vector in said base codebook B.

30

5. A computer system for creating optimal codebooks, said system comprising:

- a) a general purpose computer;
b) memory connected to said general purpose computer;
c) a compression module resident in said memory b); wherein said
5 compression module comprises;
i) an initialization module for calculating initial values for a base
codebook B, said initialization values comprising source vectors from a block
of data;
ii) a nearest neighbour module, said nearest neighbour module
10 accepting said base codebook B as input from said initialization module and
assigning each source vector in said data block to a vector in said base
codebook B; and calculating a codebook C based upon the values in base
codebook B;
iii) a centroid module, said centroid module recalculating the
15 values contained in base codebook B to improve convergence;
iv) a convergence module, said convergence module determining if
said centroid module has converged the values in said base codebook B; and
d) a graphics engine connected to said compression module.
- 20 6. The system of claim 5 wherein said initialization module initializes said base
codebook B by randomly selecting source vectors from said block of data, without
replacement.
7. The system of claim 6 wherein should said base codebook B, become singular,
25 reinitializing said codebook B.
8. The system of claim 7 wherein said reinitialization comprises selecting an
outlying source vector from said block of data.

9. The system of claim 5 wherein said nearest neighbour module creates codebook C by using the formula: $C = B * W$.
10. The system of claim 5 wherein said centroid module recalculates the values in base codebook B by using the formula: $B = SW^T * \text{inverse}(WNW^T)$,
5 wherein S is a matrix where each column contains the sum of source vectors currently assigned to each base codevector in B and N is a diagonal matrix with the number of source vectors which are the closest to each base codevector in B appearing on the diagonal.
- 10
11. The system of claim 5 wherein said convergence module returns control to said nearest neighbour module should the values in base codebook B not be converged.
- 15 12. A system for creating an optimal codebook; said system comprising:
- a) means for initializing a base codebook B;
 - b) neighbour means for creating a codebook C from said base codebook B;
 - 20 c) means for recalculating said base codebook B; and
 - d) means for determining if the contents of said base codebook B have converged.
13. The system of claim 12 wherein said means for initializing comprises
25 randomly selecting source vectors from a block of data, without replacement.
14. The system of claim 13 wherein if said base codebook B is singular, invoking means to reinitialize said base codebook B.

15. The system of claim 14 wherein said means to reinitialize comprises selecting an outlying source code vector from within a block of data and inserting said outlying source code vector in said base codebook B.
- 5 16. The system of claim 12 wherein said means for creating a codebook comprises finding the nearest neighbour for each source vector in a block of data and associating said each source vector with a base codevector in said base codebook B.
- 10 17. The system of claim 12 wherein said codebook C is calculated utilizing the formula, $C = B * W$.
- 15 18. The system of claim 12 wherein said means for recalculating said base codebook B comprises updating said base codebook B utilizing the formula: $B = S W^T * \text{inverse}(W N W^T)$, wherein S is a matrix where each column contains the sum of source vectors currently assigned to each base codevector in B and N is a diagonal matrix with the number of source vectors which are the closest to each base codevector in B appearing on the diagonal.
- 20 19. The system of claim 12 wherein if said means for determining if the contents of base codebook B have converged, determine that the values contained in base codebook B have not converged, returning control of said system to said nearest neighbour means.
- 25 20. A computer readable medium containing instructions for creating an optimal codebook; said instructions comprising the steps of:
- a) initializing a base codebook B;
 - b) finding the nearest neighbour for each source vector in a block of data and associating said each source vector with a codevector in said base codebook B;
 - 30 c) creating a codebook C based upon the formula, $C = B * W$;

- d) updating base codebook B utilizing the following formula:
$$B = SW^T * \text{inverse}(WNW^T)$$
, wherein S is a matrix where each column contains the sum of source vectors currently assigned to each base codevector in B and N is a diagonal matrix with the number of source vectors which are the closest to each base codevector in B appearing on the diagonal; and
- 5 e) repeating steps b) to d) until the values in base codebook B have converged.
21. The medium of claim 20, wherein at step a) said initializing comprises
- 10 randomly selecting source vectors from a block of data, without replacement.
22. The medium of claim 21 wherein if said base codebook B is singular, reinitializing said base codebook B.
- 15 23. The medium of claim 22 wherein said reinitializing comprises selecting an outlying source code vector from within said block of data and inserting said outlying source code vector in said base codebook B.
24. The system of claim 5 wherein if said graphics engine does not provide a
- 20 conventional interface, said compression module providing to said graphics engine non conventional 3D, 2D and 1D image data.
25. The system of claim 24 wherein said non conventional 3D data comprises: four base codevectors, each comprising 16 bits, and sixty four, three bit indices.
- 25 26. The system of claim 24 wherein said non conventional 2D data comprises: three base codevectors, each comprising 21 bits, and sixty four, three bit indices.
27. The system of claim 24 wherein said non conventional 1D data comprises: two
- 30 base codevectors, each comprising 16 bits and thirty two, 7 bit indices.